ST MICROELECTRONICS BLDC DRIVE DEMONSTRATOR SOFTWARE PRE-DEVELOPMENT

Jiri Ctibor

Doctoral Degree Programme (3), FEEC BUT

E-mail: jiri.ctibor@vutbr.cz

Ivo Pazdera

E-mail: pazderai@feec.vutbr.cz

Abstract: In this article is presented the own application structure of a BLDC motor control demonstrator. This demonstrator is made for the desirables of R3-PowerUP project advertised by the ST Microelectronics. Therefore, mostly ST devices are used, for example, STM32 microcontroller. The article describes the application structure, used drivers and peripherals. As a result, phase currents and speed estimation from rotary encoder are shown in the form of graphics view of variables obtained directly from the microcontroller.

Keywords: BLDC, motor control, STM32, R3-PowerUP

1 INTRODUCTION

The project R3-PowerUP is a European project of up to thirty company and institution contributors. The submitter of this project is ST Microelectronics. The result will be a new generation of 300 mm silicon wafer pilot line for smart power technology in Europe. This pilot line will enable the sub 100 nm Smart Power processes, starting from the 90 nm BCD10 [1]. This article will describe mainly preliminary motor control software development on a development kit serving as an exemplar for the future demonstrator with own hardware and finally for the final demonstrator which will use the new BCD integrated circuit combining power and control stage as one of a result of project R3-PowerUP. The selected development kit for preliminary software evaluation is NUCLEO-F446RE containing 32-bit ARM MCU. This MCU disposes of Cortex-M3 core designed for such real-time signals processing applications. The two significant features for selecting this core was a floating point unit and DSP instructions set, suitable right for real-time signal processing. The Nucleo board is connected to the power stage which is Nucleo expansion board X-NUCLEO-IHM07M1 – an expansion kit for developing motor control applications containing three-phase H-bridge ST L6230 and current sensing circuits. The ST L6230 could be an exemplar for the newly developed BCD integrated circuit.

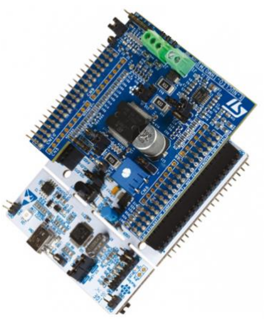

Figure 1: Development kit

Figure 2: Maxon BLDC motor

2 BCD TECHNOLOGY

The BCD technology was invented by ST Microelectronics in 1984. It is an integrated circuit fabrication process that combines three different technologies: bipolar, CMOS and DMOS all on the same silicon wafer. This technology enables manufacturers today to produce smart power ICs which combines intelligent and power stage on one chip. It could be for example motor control ICs, switch mode power supplies ICs which are widely used today in electronics. Combination of the following three technologies brings their advantages into one device. Bipolar technology is used to high precision bipolar circuits, CMOS technology is well suited to high-density analog and digital functions and DMOS for power stages [2].

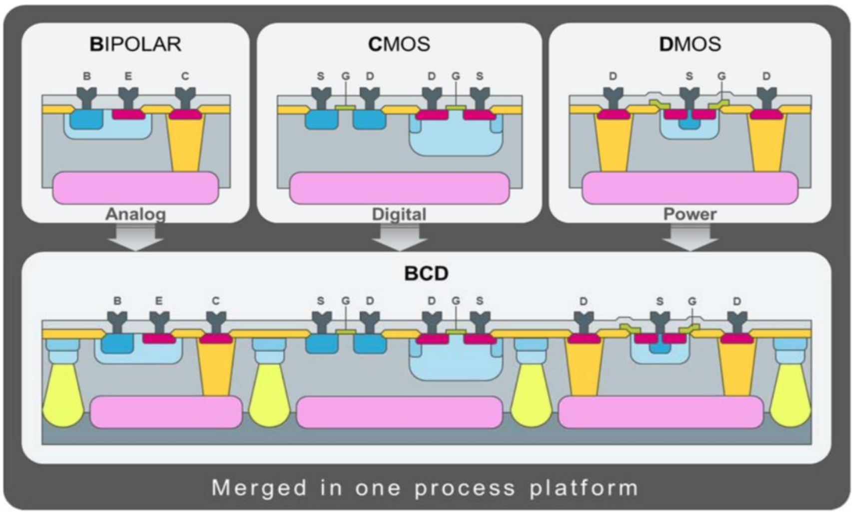

Figure 3: The BCD technology description [3]

3 DEMONSTRATOR REQUIREMENTS

Demonstrator's scope

- Small size BLDC motor controller with high power/size ratio for industrial or automotive applications.

Features

- Output power up to 2 kW at a small size with DC – link voltage 48 V.
- Motor speed range 0 – 4000 rpm.
- No need of active cooling thanks to the utilization of power MOS-FETs with ultra-low $R_{DS,on}$.
- Design for a motor with trapezoidal back EMF waveform, i.e. rectangular phase current waveform.
- Sensor-less control algorithm running on STM32 MCU.
- Cascade control structure.

4 SOFTWARE STRUCTURE

The basic software structure overview from applications layers point of view could be explained by the following picture Figure 4.

The lowest layer used to be called Low-Level peripherals layer and forms the physical connection between hardware (transistors etc.) and software. For motor control, there has to be a PWM periphery, in STM32 microcontroller is made from the Advanced control timer. To control a BLDC motor, it is required (before any sensor-less algorithm is utilized) to know the rotary position of the rotor. For this reason, there are three hall sensors mounted on the stator, from which then can the actual position in one of the six sectors be determined. The instances, in which the commutation from one sector to another happens is determined by a general-purpose timer, set up in the Hall XOR mode. Because there is a requirement for the advanced current control, also ADC converters to current sensing are utilized. For the simplicity and because this STM32 microcontroller comes with three physical ADC converters, all three converters are utilized. The last used peripheral is UART to serial communication with a superordinate control system, or for the human-machine interface. This bus can be finally substituted by some industrial standard bus like CAN, LIN or Ethernet.

The mid layer is called Driver layer and contains peripheral control. Because for creating the SW project the STM CubeMX is used which comes also with peripheral control libraries, the so-called Low-Level (LL) libraries are used because of its universality and better procession speed compared to Hardware Abstraction Libraries (HAL). For the digital signal processing, the original ARM CMSIS-DPS library is used for now. Especially the low pass filters and PID controllers are exploitable from this library. But the offered controllers have few disadvantages for example abstention of anti-windup protection, output saturation so they may be finally rewritten. The last part of driver layer is user software functions, mainly the commutation table saying in which sector, what transistors should be switched to generate the required torque, and phase PWM duty cycles counting functions (SVM algorithms).

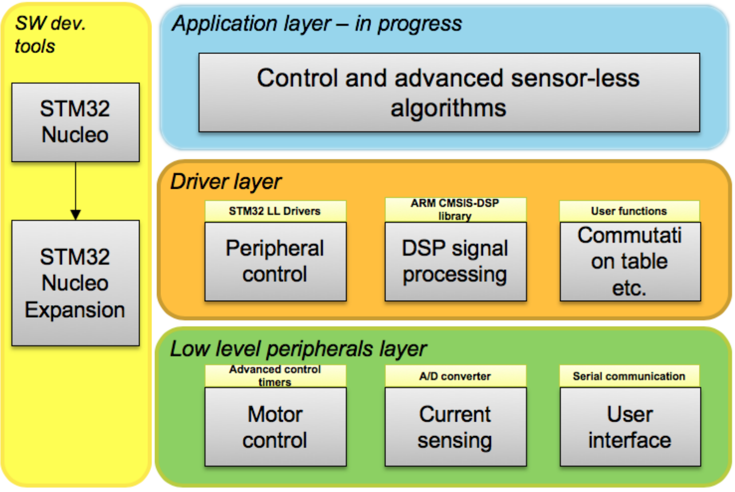

Figure 4: Application layers separation

These two layers are basically written and optimized now. The superior Application layer is in progress, the sensor-less algorithms could be included there in the future.

The following Figure 5 depictures detailed system block diagram. It is divided into three columns. The very left column contains all used hardware. The base of this section is a BLDC motor with built-in hall sensors. The motor is connected through the shunt resistors for current sensing to the three-phase low voltage converter chip. For software developing purposes especially for the sensor-less algorithm also a quadrature encoder is employed for speed or position estimation and comparing to the algorithm outputs. The last thing in the hardware section is analog circuits for signal conditioning of the signals from shunt resistors.

The middle column contains also hardware but inside the microcontroller, so this section is therefore called peripheral section. The timer TIM1 is an advanced control timer configured in PWM mode. The counter is set into PWM 2 mode and Up/Down counting mode. One live phase then got duty cycle *s* and second live phase to 1-*s*. The third dead phase is left turned off by disable input of the converter chip. These settings lead to the center-aligned PWM signals and so-called unipolar PWM signal generation.

Timer TIM2 is the interfacing timer for the hall sensors. This timer is configured to the special *hall xor* mode as figured out in Figure 1. Basically, all three hall signals are guided through logical "or" so every commutation instant of each of the hall sensor cause edge on the output and therefore triggers the commutation routine *get sector* followed by *select phase* routine selecting the current phase according to commutation table.

Timer TIM4 is the interfacing timer for quadrature encoder. It is configured in the special encoder mode, which is very easy to use. The timer is counting up or down by themselves according to the rotation direction without any software control. Estimated speed is then calculated as a difference of the value of the TIM4 between two periodical time spans.

Timer TIM10 is used as a periodic interrupt timer with frequency 1 kHz, which triggers the speed loop counting routine.

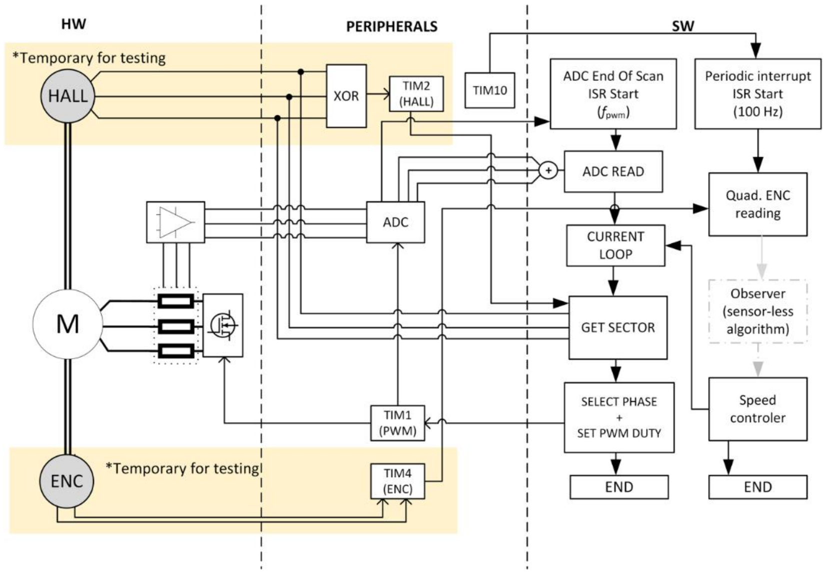

Figure 5: Detailed system block structure from the software point of view

Finally, there are also three ADC converters in this microcontroller so there is no need to count the phase current from only two currents and it is possible to sample all three phase currents in one instance. The ADC converters are triggered by the TIM1 (PWM), allowing the currents to be sampled every time in the same instants of a PWM period. This is very important, because of the sawtooth shape of the current due to the induction of the motor winding.

The last right column is pure software section. There are described two periodic routines. The first one is the current loop routine. This routine begins, when the ADC converter triggers the End of Scan interrupt. Then the currents are read, and some signal conditioning and filtering are done in *ADC read* routine. It is followed by the *current loop* routine, where the PID calculation is performed. As it was proposed, for now, the original ARM CMSIS-DPS PID library is used, until it fulfills our requirements. Controller routine is modified, saturation and anti-windup protection have been added externally. When the required duty cycles are known, the algorithm needs to know in which sector the rotor actually is, this is detected in *get sector* routine, which is basically GPIO reading of hall output signals. This routine is also called asynchronously from the TIM2 event generated by the hall sensors. The last subroutine of current loop routine is selecting the current phase from the commutation table and writing the duty cycles into the PWM (TIM1).

Second important software loop is speed loop. This loop is triggered periodically (1 kHz) by the TIM10 because it is required to read the actual TIM4 value and to know the difference of this value, which then expresses the actual speed of the rotor. This section will be substituted by some kind of sensor-less algorithm in the future. And finally, it is followed by the PID speed loop calculation, which is the same as the current loop PID controller [4].

5 RESULTS AND MEASUREMENTS

In this article mostly the low level and driver layer have been discussed. There is also some progress with current and speed control, but the controllers are not well tuned now, therefore, there are no results covering the controller's setup and verification. It will be done in some future articles. Following figures Figure 6 and Figure 7 are summarizing results of this developing phase. There are depicted the raw and recalculated currents and shaft rotation speed in an instance when the full duty cycle has been applied to the stopped motor. These waveforms are live values of variables taken directly from the microcontroller by software development tools SEGGER J-Scope.

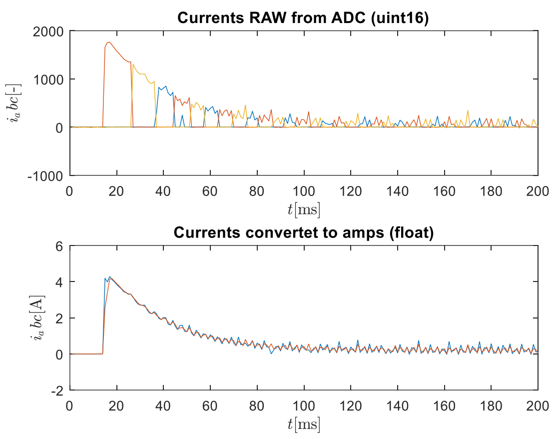

Figure 6: Phase currents

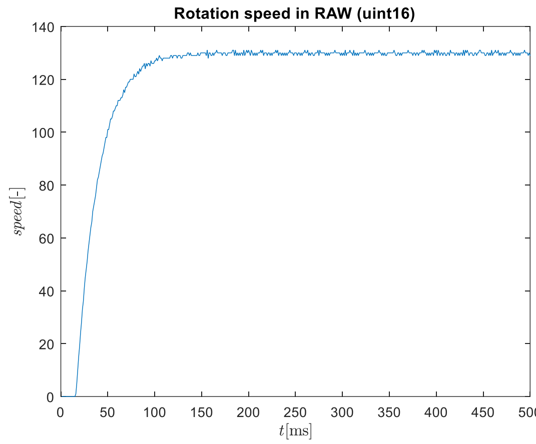

Figure 7: Shaft rotation speed

ACKNOWLEDGEMENT

This research was carried out under the project 737417-2 R3-PowerUP 300mm Pilot Line for Smart Power and Power Discretes financed by H2020-ECSEL-2016-2-IA-two-stage and under the project LQ1601 CEITEC 2020 financed by MEYS in the National Sustainability Programme. This research also acknowledges to the Centre for Research and Utilization of Renewable Energy (CVVOZE). Authors gratefully acknowledge financial support from the Ministry of Education, Youth and Sports of the Czech Republic under NPU I programme (project No. LO1210) and BUT specific research programme (project No. FEKT-S-17-4374).

REFERENCES

[1] ECSEL 2016-2-IA-737417: R3-PowerUP: 300mm Pilot Line for Smart Power and Power Discretes. STMicroelectronics Italy, 2017. URL: https://r3powerup.eu/

[2] A. Baiocchi, "New developments in mixed bipolar/CMOS/DMOS technology for intelligent power applications," *IEE Colloquium on Integrated Power Devices*, London, UK, 1991, pp. 2/1-2/4. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=180811&isnumber=4540

[3] *BCD Technology: Sense & Power and Automotive Technology R&D*. Italy, 2017. URL: https://mycmp.fr/IMG/pdf/bcd8s-soi_technology_overview-2.pdf

[4] *RM0390 Reference manual: STM32F446xx advanced Arm®-based 32-bit MCUs*. 2018. URL: https://www.st.com/content/ccc/resource/technical/document/reference_manual/4d/ed/bc/89/b5/70/40/dc/DM00135183.pdf/files/DM00135183.pdf/jcr:content/translations/en.DM00135183.pdf